MXNET COMPUTER VISION AND NATURAL LANGUAGE PROCESSING MODELS ACCELERATED WITH NVIDIA TENSORCORES

Cyrus Vahid (Amazon), Przemyslaw Tredak (NVIDIA), 3.20.2019
Apache MXNet (incubator)
GOALS

- Developer Productivity
- Inference Efficiency
- Training Efficiency
GOALS

Developer Productivity

Interoperability

Inference Efficiency

Training Efficiency
Developer Productivity
MULTI-LANGUAGE SUPPORT

Java
Perl
Julia
Clojure
Python
Scala
C++
R

Frontend

Backend

C++
WHY GLUON

- Simple, Easy-to-Understand Code
- Flexible, Imperative Structure
- Dynamic Graphs
- High Performance
NETWORK DEFINITION IN GLUON

```
net = gluon.nn.HybridSequential()

with net.name_scope():
    net.add(gluon.nn.Dense(units=64, activation='relu'))
    net.add(gluon.nn.Dense(units=10))

softmax_cross_entropy = gluon.loss.SoftmaxCrossEntropyLoss()

net.initialize(mx.init.Xavier(magnitude=2.24), ctx=ctx, force_reinit=True)

trainer = gluon.Trainer(net.collect_params(), 'sgd', {'learning_rate': 0.02})
```
for e in range(10):

cumulative_loss = 0

for i, (data, label) in enumerate(train_data):

    data = data.as_in_context(model_ctx).reshape((-1, 784))

    label = label.as_in_context(model_ctx)

    with autograd.record():

        output = net(data)

        loss = softmax_cross_entropy(output, label)

        loss.backward()

    trainer.step(data.shape[0])
HYBRIDIZE

```python
net.hybridize(static_alloc=True, static_shape=True)
```

- NUM_GPU: 1, NUM_WORKER: 29,
- BATCH_SIZE_PER_GPU: 64.0,
- TYPECAST: `<class 'numpy.float32'>`
- SYMBOLIC: False
- Samples/Sec: 1600
- epoch time: 32:00

- NUM_GPU: 1, NUM_WORKER: 29,
- BATCH_SIZE_PER_GPU: 64.0,
- TYPECAST: `<class 'numpy.float32'>`
- SYMBOLIC: True
- Samples/Sec: 3200
- epoch time: 16:00
GLUONCV: A DEEP LEARNING TOOLKIT FOR COMPUTER VISION

https://gluon-cv.mxnet.io
GLUONCV.MODELZOO

• 50+ Pre-trained models, with training scripts, datasets, tutorials

```python
net = get_model('resnet50_v2',
    classes=100,
    pretrained=False)
```

• For a complete list of the pre-trained models please refer to:
  https://gluon-cv.mxnet.io/api/model_zoo.html
GLUONCV PRE-TRAINED MODELS
PRETRAINED MODELS

```python
net = get_model('resnet50_v2',
    classes=100,
    pretrained=True)

• Transfer Learning
  ...pretrained=True)

• Inference
  pred = net(img.expand_dims(axis=0))
  ind = nd.argmax(pred, axis=1).astype('int')
  nd.softmax(pred)[0][ind].asscalar()
```
GLUONCV EXAMPLE CODE
CHICK-FIL-A KEEPS WAFFLE FRIES FRESH

- Track waffle fry freshness
- Identify fries that have exceeded hold time
- Gluon Computer vision model for object detection and tracking
- A team of students with no ML expertise
- 12 months from no ML knowledge to completion
GluonNLP
FEATURES

- 300+ word embedding pre-trained models
- 5 language models
- Neural Machine Translation (Google NMT, Transformer)
- Flexible data pipeline tools
- Public datasets
- NLP examples, e.g. sentiment analysis
**GLUONNLP APIS**

**gluonnlp.data**: Build efficient data pipelines for NLP tasks

**gluonnlp.vocab**: Provides text data numericalization and the subword functionality

**gluonnlp.model**: Train or load state-of-the-arts models for common NLP tasks

**gluonnlp.embedding**: Train or load state-of-the-arts embeddings for common NLP tasks
BUCKETING

Average Padding = 11.7

Data loading slow and memory inefficient

GluonNLP data bucketing fast and memory efficient

Average Padding = 3.7
• Our implementation: BLEU 26.22 on IWSLT2015, 10 epochs, Beam Size=10
• Tensorflow/nmt: BLEU 26.10 on IWSLT2015, Beam Size=10
NMT - TRANSFORMER

Encoder

• 6 layers of self-attention+ffn

Decoder

• 6 layers of masked self-attention and
• output of encoder + ffn

• Our implementation: BLEU 26.81 on WMT2014en_de, 40 epochs
• Tensorflow/t2t: BLEU 26.55 on WMT2014en_de
EMBEDDING

Language model, machine translation, QA, Dialog System, etc.

Word2vec, Fasttext, Glove, etc

Word Embedding, Sentence Embedding, Paragraph embedding etc.

Language Embedding

Recommendation
Information Retrieval
Advertising, etc.

Graph mining etc.

LINE, Deepwalk, CNN embedding

Network embedding, Subgraph embedding

Graph Embedding

Image classification, Image detection, SSD, etc

Faster R-CNN, etc

CNN embedding

Image Embedding

Embedding

Network mining etc.
In GluonNLP, we provide

- High-level APIs
  - gluonlp.data, gluonlp.model, gluonlp.embedding
- Low-Level APIs
  - gluonlp.data.batchify, gluonlp.model.StandardRNN

*Designed for practitioners: researchers and engineers*
Amazon SageMaker Neo
AMAZON SAGEMAKER NEO
TRAIN ONCE, RUN ANYWHERE WITH 2X THE PERFORMANCE

Get accuracy and performance

Automatic optimization

Broad framework support

Broad hardware support

KEY FEATURES

Open-source Neo-AI device runtime and compiler under the Apache software license; 1/10th the size of original frameworks
TRAIN ONCE, RUN ANYWHERE WITH 2X THE PERFORMANCE

https://amzn.to/2Hlj3ws
TENSORCORES AND MIXED PRECISION

Starting with Volta, NVIDIA GPUs feature TensorCores.

They greatly speed up matrix multiplication.

Using them requires mixed precision.
MIXED PRECISION RECIPE

Theory

- Cast input to FP16, cast back to FP32 before the softmax
- Keep “master copy” of the weights in FP32
- Scale the loss to keep gradients in FP16 dynamic range
MIXED PRECISION RECIPE

In practice

- Cast input to FP16, cast back to FP32 before the softmax
  - What about Norm, Mean, etc.?
- Keep “master copy” of the weights in FP32
  - optimizer.multi_precision=True
- Scale the loss to keep gradients in FP16 dynamic range
  - What should the loss scale be? How to make it dynamic?
AMP: AUTOMATIC MIXED PRECISION

- Automatic casting of the model
  - Convolution, FullyConnected -> FP16
  - Norm, Mean, SoftMax, etc. -> FP32
  - Add, Mul etc. -> Cast to widest type
- Utilities for dynamic loss scaling
AMP: AUTOMATIC MIXED PRECISION

- Automatic casting of the model
  - Convolution, FullyConnected -> FP16
  - Norm, Mean, SoftMax, etc. -> FP32
  - Add, Mul etc. -> Cast to widest type
- Utilities for dynamic loss scaling
AMP: AUTOMATIC MIXED PRECISION

- Automatic casting of the model
  - Convolution, FullyConnected -> FP16
  - Norm, Mean, SoftMax, etc. -> FP32
  - Add, Mul etc. -> Cast to widest type
- Utilities for dynamic loss scaling
AMP: AUTOMATIC MIXED PRECISION

- Automatic casting of the model
  - Convolution, FullyConnected -> FP16
  - Norm, Mean, SoftMax, etc. -> FP32
  - Add, Mul etc. -> Cast to widest type
- Utilities for dynamic loss scaling
AMP: AUTOMATIC MIXED PRECISION

- Automatic casting of the model
  - Convolution, FullyConnected -> FP16
  - Norm, Mean, SoftMax, etc. -> FP32
  - Add, Mul etc. -> Cast to widest type
- Utilities for dynamic loss scaling
net = get_network()
trainer = mx.gluon.Trainer(...)

with autograd.record(True):
    out = net(data)
    l = loss(out, label)

    autograd.backward(loss)
MIXED PRECISION RECIPE

AMP

```python
amp.init()
net = get_network()
trainer = mx.gluon.Trainer(...)
amp.init_trainer(trainer)
with autograd.record(True):
    out = net(data)
    l = loss(out, label)
    with amp.loss_scale(loss, trainer) as scaled_loss:
        autograd.backward(scaled_loss)
```
PERFORMANCE - CLASSIFICATION

Speedup when using AMP (single GPU, same batch size)
PERFORMANCE - DETECTION

Speedup when using AMP (single GPU, same batch size)
PERFORMANCE - SEGMENTATION

Speedup when using AMP (single GPU, same batch size)
TRAINING OPTIMIZATION

MLPerf
TRAINING OPTIMIZATION

Holistic approach

- GPU kernel optimization is only one element of speeding up training

- Other potential bottlenecks
  - Data loading and augmentation
  - Operator launch overhead
TRAINING OPTIMIZATION

Data pipeline and augmentation
TRAINING OPTIMIZATION

CPU optimization

Operator – big batch size

Launch  Wait  Update

GPU kernel

Operator – small batch size

Launch  Wait  Update

GPU kernel

- constant overheads
TRAINING OPTIMIZATION

CPU optimization

Operator bulk

GPU kernel 1  GPU kernel 2  GPU kernel 3  GPU kernel 4

Launch 1  Launch 2  Launch 3  Launch 4  Wait  Update

0% 10% 20% 30% 40%

BS 256
BS 32
BS 8

Speedup of bulking for different batch sizes
# NVIDIA LED DGX SESSIONS AT GTC 2019

NOTE: For details on all DGX-related sessions, visit: [GTC site](#) and search for “DGX” or look-up session ID

<table>
<thead>
<tr>
<th>Session #, Date/Time</th>
<th>Location</th>
<th>Session Name</th>
<th>Product Featured</th>
</tr>
</thead>
<tbody>
<tr>
<td>S9483 Mon 3/18, 9am</td>
<td>Marriott Hotel Ballroom 3</td>
<td>Creating AI Workgroups Within The Enterprise: Developers Share Their Best Practices</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Markus Weber and Michael Balint (DGX) + Customer <strong>Subtle Medical</strong></td>
<td></td>
</tr>
<tr>
<td>S9120 Tue 3/19, 10am</td>
<td>Convention Center Room 212B</td>
<td>How to Accelerate and Scale A.I. Deployment with Proven Architecture Designs</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Charlie Boyle (DGX Product Management) and Ludwig Gamache (Head of IT <strong>ElementAI</strong>)</td>
<td></td>
</tr>
<tr>
<td>S9121 Tue 3/19, 2pm</td>
<td>Marriott Hotel Ballroom 3</td>
<td>Deep Learning Implementers Panel: Experts Discuss The Keys to Their Success</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Tony Paikeday (DGX), Zachary Hanif (<strong>Capital One</strong>), Enhao Gong (<strong>Subtle Medical</strong>), Norman Mueller (<strong>BMW</strong>)</td>
<td></td>
</tr>
<tr>
<td>S9500 Wed 3/20, 9am</td>
<td>Convention Center Room 212B</td>
<td>Latest Deep Learning Framework Container Optimizations</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Michael O’Connor and Joey Conway (Deep Learning Software)</td>
<td></td>
</tr>
<tr>
<td>S9334 Wed 3/20, 10am</td>
<td>Convention Center Room 212B</td>
<td>AI Infrastructure: Lessons Learned from NVIDIA DGX POD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Darrin Johnson + Jacci Cenci (DGX Tech Marketing), Andrew Bull + Sumit Kumar (Solution Architects)</td>
<td></td>
</tr>
<tr>
<td>S9893 Wed 3/20, 1pm</td>
<td>Convention Center Room 212B</td>
<td>KVM GPU Virtual Machines: Maximizing Performance and Utilization on <strong>DGX-2</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Anish Gupta, Software Engineer</td>
<td></td>
</tr>
<tr>
<td>S9241 Wed 3/20, 1pm</td>
<td>Convention Center Room 220C</td>
<td>All You Need to Know about Programming NVIDIA’s <strong>DGX-2</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Lars Nyland and Stephen Jones, Software Engineers</td>
<td></td>
</tr>
<tr>
<td>S91003 Wed 3/20, 2pm</td>
<td>Convention Center Room 210A</td>
<td>MXNet Computer Vision and Natural Language Processing Models Accelerated with NVIDIA TensorCores</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Przemyslaw Tredak (DevTech Engineer) and Cyrus Vahid (Principle Evangelist <strong>AWS Deep Engine</strong>)</td>
<td></td>
</tr>
<tr>
<td>Session #, Date/Time</td>
<td>Location</td>
<td>Session Name</td>
<td>Product Featured</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------</td>
<td>--------------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| S9292 Mon 3/18, 10am | SJ Convention Center Room 212B | Red Hat and the NVIDIA DGX: Tried, Tested, Trusted  
  - Jeremy Eder, Software Engineer, Red Hat, and Charlie Boyle (DGX Product Management) | DGX-1 |
| S9164 Tue 3/19, 9am | Hilton Hotel Market Room | Advanced Weather Information Recall with DGX-2  
  - Tomohiro Ishibashi, Director, Weather News, and Shigehisa Omatsu, CEO, dAlgnosis, Inc. | DGX-2 |
| S9983 Tue 3/19, 9am | Marriott Hotel Ballroom 5 | Edge to Core: A Meta Study of Data Complexity in AI  
  - James Coomer, Senior Vice President Products, DDN | DDN |
| S9325 Tue 3/19, 10am | SJ Convention Center Room 220B | Machine Learning in Action within a Large Regional Healthcare System  
  - Brandon Fornwalt, Associate Professor, Geisinger | DDN |
| S9373 Tue 3/19, 3pm | Marriott Hotel Ballroom 2 | TPC-H Benchmark on DGX-2: A New Paradigm for OLAP and Decision Support  
  - Richard Heyns, CEO, and Piotr Kowalski, Senior Engineer, Brytlyt | DDN |
| S9417 Wed 3/20, 3pm | SJ Convention Center Room 211B | Molecular Generative VAEs: Parallelization, Optimization, and Latent Space Analysis on DGX-1  
  - Ellen Du and Joey Storer, Research Scientists, Dow Chemical Company | DGX-1 |
| S9469 Wed 3/20, 4pm | SJ Convention Center Room 231 | MATLAB and NVIDIA Docker: A Complete AI Solution, Where You Need It, in an Instant  
  - Jos Martin and Joss Knight, Engineering, MathWorks | DGX-2 |
| S9892 Wed 3/20, 4pm | SJ Convention Center Room 220A | Deep Learning for Autonomous Driving at BMW  
  - Alexander Frickenstein, PhD Candidate, BMW | DGX-2 |
| S9406 Thu 3/21, 3pm | SJ Convention Center Room 212B | Hybrid Cloud for Flexible GPU Resource Planning and Orchestration  
  - Jeongkyu Shin, CEO and Joongi Kim, CTO, Lablup, Inc. | DGX-2 |